World Applied Sciences Journal 16 (Special Issue of Applied Math): 57-63, 2012
ISSN 1818-4952

An Optimal Approach for Minimization of Total Costs in the Multi-mode Resource-constrained Project Scheduling Problem

Mohammad Khalilzadeh

Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

Abstract: In this paper, we introduce a new multimode project scheduling problem, taking into account finish-to-start precedence relations among project activities as well as renewable and nonrenewable resources constraints. We suppose renewable resources are hired and not accessible in all time intervals of a project. In other words, there is a presumed release date as well as a mandated due date for each renewable resource type such that no resource can be available before its release date. However the resources are allowed to be executed later than their due dates by remitting penalty costs based on the resource type. The objective is to minimize the entire costs of renewable and nonrenewable resources, called multi-mode resource-constrained project scheduling problem, minimization of total weighted resource tardiness penalty cost (MRCPSP-TWRTPC) where nonrenewable resource costs depend on the activity modes. In this paper, we present a branch and bound algorithm to deal with this extended type of MRCPSP. A numerical example is presented to describe our branch and bound approach in more details. Experimental results reveal the capability of the proposed algorithm in solving the problem in question particularly for the small and medium sized test functions.

Key words: MRCPSP . renewable and nonrenewable resources . weighted tardiness . branch and bound

INTRODUCTION

In practice, the project activities can be performed in multiple possible execution modes. This yields fascinating project scheduling problem commonly known as the multi-mode resource-constrained project scheduling problem (MRCPSP). MRCPSP selects an execution mode for each project activity and determines the activity start and finish times subject to precedence constraints as well as renewable and nonrenewable resources restrictions in order to minimize the project duration. In this paper, we introduce an extended form of MRCPSP by considering renewable resources are rented and not available in all periods of time of a project. In other words, there is a mandated release date as well as a due date for each renewable resource type such that no resource can be obtainable before its release date. However the resources are permitted to be used later than their due dates by paying penalty costs based on the resource type. The objective is the minimization of the entire costs of both renewable and nonrenewable resources, called multi-mode resource-constrained project scheduling problem, minimization of total weighted resource tardiness penalty cost (MRCPSP-TWRTPC) where nonrenewable resource costs depend on the activity modes.

Considerable number of exact and heuristic methods has been presented in the literature of the MRCPSP problem. A comprehensive review on these publications can be found in Chapter 8 of Project Scheduling Handbook [3] and a survey done by Weglarz *et al*. [15].

In this paper, another project scheduling problem with the objective of resources costs minimization is taken into consideration. We try to study an extended and more practical model by considering the costs of both renewable and nonrenewable resources. The cost of nonrenewable resources is a function of their requirements determined by activities modes. Also, renewable resources are available in pre-determined time periods specified by their release times and due dates. No renewable resources can be available before their release times however using these resources after their due dates will yield penalty costs. This is motivated by the case renewable resources are rented for specified periods that using them after those periods causes tardiness penalty cost. As the renting cost is fixed for the periods, there is no need to enter it in objective function and only tardiness penalty cost can be considered for these resources.

Resources tardiness has been only examined by Ranjbar *et al*. [13] taking into account unary renewable

Corresponding Author: Mohammad Khalilzadeh, Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

resources. However, they only considered renewable resources and single mode activities. This problem called resource-constrained project scheduling problem, minimization of total weighted resource tardiness penalty cost (RCPSP-TWRTPC), was introduced as an extension of resource-constrained project scheduling problem (RCPSP). They developed an exact algorithm based on branch and bound approach to deal with this problem.

Therefore, our proposed problem is a generalization of the problem presented by Ranjbar *et al*. [13] with more realistic viewpoint of resources costs. We call this problem multi-mode resource-constrained project scheduling problem, minimization of total weighted resource tardiness penalty cost (MRCPSP-TWRTPC).

The problem described here is an extension of the RCPSP problem and considering the NP-hardness of RCPSP ([1, 2], the MRCPSP-TWRTPC problem is NP-hard as well. Among several exact and non-exact algorithms presented in the literature for solving MRCPSP we can point to branch and cut method proposed by Heilmann [5] and branch and bound method developed by Zhu *et al*. [16] as two of the most powerful exact methods. Also Lova *et al*. [8] proposed priority rules and mode selection rules to solve MRCPSP. Moreover, Genetic Algorithms suggested by Lova *et al*. [9] and Peteghem and Vanhoucke [10] and Scatter Search algorithm developed by Ranjbar *et al*. [12] can be mentioned as some of the most efficient meta-heuristic methods.

The rest of this paper is organized as the following. In the next section, MRCPSP-TWRTPC is described and mathematically formulated for the very first time. In section 3, our proposed branch and bound algorithm is presented. Section 4 is dedicated to an illustrative example to explain further our blanch-and-bound approach. The computational analyses are given in section 5. Finally, we conclude the paper in section 6.

PROBLEM MODELING AND FORMULATION

MRCPSP-TWRTPC is the generalization of MRCPSP, but instead of minimizing project duration, project cost minimization is aimed. We describe this problem in details in the following.

The objective of MRCPSP-TWRTPC is to find a feasible schedule in order to minimize the total costs of a project. Each project activity can have several execution modes in which renewable and nonrenewable resource requirements and its duration are stated as a discrete amount of unit measure. The availabilities of renewable resources are limited and their release times, deadlines and tardiness penalty costs are specified. All activities are ready at the beginning of the project, no preemption is permitted during activities executions and there are finish-to-start precedence relations among activities.

In order to mathematically formulate this problem, we consider an activity on node (AON) representation with finish-to-start precedence relations and zero time lag. Dummy activities 1 and n correspond to start and completion of the project. The list of activities is topologically numbered, i.e., each predecessor of every activity has a smaller number than the number of activity itself.

Also we define the earliest and latest start time of each activity, EST_j and LST_j, with forward and backward passes using the mode with shortest duration for each activity and assigning $LST_n= LFT_n=T$ where T is a upper bound for project duration determined by any valid method, such as the simple method of summation of the longest duration of entire project activities plus the latest ready time of the renewable resources. So each activity j can only be performed in time period $[EST_j,LST_j]$.

We define problem parameters as the following:

- n: Number of project activities
- NR: Number of nonrenewable resources
- c_k: Unit cost of nonrenewable resource k
- R: Number of renewable resources
- R_k: Renewable resource k availability
- r_k: Release time of renewable resource k
- d_k: Due date of renewable resource k
- p_k: Tardiness penalty cost of renewable resource k for each period
- M_j: Number of modes of activity j
- P_j: The set of the predecessors of activity j
- d_{jm}: Duration of activity j under mode m
- r_{jmk}: Renewable resource k requirement for executing activity j under mode m
- n_{jmk}: Nonrenewable resource k requirement for executing activity j under mode m
- EST_j: Earliest start time of activity j
- LST_j: Latest start time of activity j
- T: Project duration upper bound

We also define the decision variables as the following:

$$x_{jm\tau} = \begin{cases} 1 & \text{if activity j is started under mode m in period } \tau \\ 0 & \text{otherwise} \end{cases}$$

$$y_{k\tau} = \begin{cases} 1 & \text{if renewable resource k is used in period } \tau \\ 0 & \text{otherwise} \end{cases}$$

l_k: Renewable resource k tardiness determined by: $l_k = \max\{0, CP_k\text{-}d_k\}$ where CP_k is the release time of resource k by the project.

The mixed integer programming model for this problem can be formulated as follows:

$$Min \quad \sum_{k=1}^{NR} c_k \left(\sum_{j=1}^{n} \sum_{m=1}^{M_j} n_{jmk} \sum_{\tau=EST_j}^{LST_j} x_{jm\tau} \right) + \sum_{k=1}^{R} p_k . l_k \tag{1}$$

S.t.

$$\sum_{m=1}^{M_j} \sum_{\tau=EST_j}^{LST_j} x_{jm\tau} = 1 \qquad j = 1,2,\ldots,n \tag{2}$$

$$\sum_{m=1}^{M_i} \sum_{\tau=EST_i}^{LST_i} (\tau + d_{im}) x_{im\tau} \le \sum_{m=1}^{M_j} \sum_{\tau=EST_j}^{LST_j} \tau x_{jm\tau} \qquad j = 1,2,\ldots,n, \quad i \in P_j \tag{3}$$

$$\sum_{j=1}^{n} \sum_{m=1}^{M_j} r_{jmk} \sum_{z=\tau-d_{jm}+1}^{\tau} x_{jmz} \le R_k . y_{k\tau} \qquad k = 1,2,\ldots,R, \quad \tau = 1,2,\ldots,T \tag{4}$$

$$\sum_{\tau=1}^{r_k-1} y_{k\tau} = 0 \qquad k = 1,2,\ldots,R \tag{5}$$

$$\tau . y_{k\tau} - d_k \le l_k \qquad k = 1,2,\ldots,R \qquad \tau = d_k, d_k+1,\ldots,LST_n \tag{6}$$

$$x_{jm\tau} \in \{0,1\} \qquad j = 1,2,\ldots,n, \quad m = 1,2,\ldots,M_j \qquad \tau = EST_j,\ldots,LST_j \tag{7}$$

$$y_{k\tau} \in \{0,1\} \quad k = 1,2,\ldots,R, \quad \tau = 0,\ldots,LST_n \tag{8}$$

$$l_k \ge 0 \qquad k = 1,2,\ldots,R \tag{9}$$

In model above, objective function (1) is project cost minimization in which the first and second terms are total cost of using nonrenewable resources and total penalty cost of renewable resources tardiness respectively. Model contains 5 functional and 3 non-functional constraint sets described as the following. Constraint set (2) ensures that each activity j is started under one of its modes in its specified start time period, i.e. $[EST_j, LST_j]$. Constraint set (3) forces precedence relationship between activities. Constrains (4) regard renewable resources usage limitations. According to constraints (5), renewable resources cannot be used before their release times and their tardiness are determined by constraints (6). Finally, constraint sets (7), (8) and (9) are non-functional ones.

OUR BRANCH AND BOUND ALGORITHM

Sprecher *et al.* [14] introduced several preprocessing rules in order to reduce feasible space of MRCPSP. Later, these rules have been used in other articles such as Hartmann and Briskon [4], Lova *et al.* [9] and Peteghem and Vanhouck [10]. Considering the similarities between MRCPSP-TWRTPC and MRCPSP, we apply two of these rules to our proposed problem. One is the non-executable mode elimination rule for an activity. For a non-executable mode, the amount of the resource needed for executing the activity is more than the resource availability. Another method is inefficient mode elimination method. A given mode is inefficient for an activity if there is another mode for which the activity duration is less and that activity can be accomplished with less total amount of both renewable and nonrenewable resources. Therefore, activities modes are analyzed one by one and non-executable and inefficient modes are deleted.

The idea of developed branch and bound algorithm for the problem is based on the generation of feasible activity list which adds activities to the schedule until a feasible accomplished schedule is generated. At each iteration the next activity in the priority list is chosen and the first possible starting time is assigned for that activity such that no precedence constraint is violated.

A partial activity list (AL) is completed along the branching tree. In each step, one activity is chosen for the AL from the set of available activities that have not been selected yet, but their predecessors have all been. The selected activity is placed at the end of current partial AL. As activities are considered as multiple modes in our problem, having selected each activity i for a partial AL, we add m_i nodes to the branching tree where m_i is the number of possible modes in which activity i can be accomplished. So in each generated node, activity i is executed under the related assigned mode and in this way, mode assignment is completed along with AL generation.

A lower bound for the related cost of each node is determined as follows. The related costs of nonrenewable resources are calculated for activities which their modes have been assigned and for the rest of activities, modes with the least related cost of nonrenewable resources are chosen and in this way a lower bound for the total cost of nonrenewable resources is determined. Regarding to the related costs of renewable resources, the minimum tardiness is to be determined for each resource and a schedule is generated based on the current partial AL and mode assignment. We generate a schedule to get a lower bound for resource tardiness so that the finish times of all activities are earlier or equal to any final feasible

1. Do preprocessing
2. Generate the initial node with the first dummy activity and select it for branching
3. Branch and fathom the selected node
4. Fathom and assess each new node that contains all activities and update current upper bound if necessary
5. If upper bound was updated in step 4:
    a. Check for fathoming all nodes that are not fathomed yet

   Else:
    b. Check for fathoming new nodes generated by last branching
6. If all nodes are fathomed:
    a. Report the best feasible solution achieved and stop

   Else:
    a. Choose a node for branching from the set of open nodes
    b. Continue from step 3

Fig. 1: The Pseudo-code of the Branch and bound algorithm

schedule. So we determine the finish times of activities as follows.

For activities present in current partial AL whose modes have been assigned, we use serial schedule generation for scheduling them and determining their start and finish times. As activity lists are generated under precedence feasible condition, it is possible to transform a partial AL to a feasible schedule.

For activities not listed in AL, we use forward pass method to determine earliest possible start times and finish times. We briefly describe our proposed steps as follows:

- We first determine the earliest start time of activity in the way that an activity cannot start earlier than the earliest finish time of all of its predecessors.
- In the second step, we determine earliest start time of the activity under each of its modes as the following: According to the usage of renewable resources for each activity, a lower bound on the start time and finish time of activity is achievable. For each activity, the earliest start time under any of its modes cannot be sooner than the release time of all resources that activity needs under that mode. So the earliest start time of activity under each mode cannot be less than the maximum of release time of renewable resources it needs under that mode and the earliest start time of activity determined considering the finish time of its predecessors.
- The earliest finish time of each activity under each mode is simply determined by adding the duration of the activity under that mode to its earliest start time. Subsequently, in the next step, earliest finish time of the activity under each mode is determined. Since we try to lessen the finish times of activities, the mode with the least earliest finish time is chosen and the earliest start time and finish time of the activity is set.

Having determined the earliest finish time of all activities, we can calculate the renewable resources tardiness. For activities present in partial AL, the usage of renewable resources during time can be determined based on the schedule. In order to get a lower bound on the renewable resources tardiness, a renewable resource is needed for each activity not listed in AL only if it is needed under all modes of the activity.

Therefore, a lower bound on the objective function value of each node is found by adding the lower bound of nonrenewable resources cost and renewable resources cost. It is obvious that nodes with lower bounds more than the current upper bound for objective function value are fathomed. Also, the upper bound for objective function value is updated based on the feasible nodes on the last level of branching tree.

It is clear that a tight initial upper bound for the optimum value of the objective function hastens fathoming nodes and speed up the whole process. In order to increase the speed of the procedure, the objective function value of the final solution obtained by an efficient meta-heuristic algorithm is used as an initial upper bound for our proposed branch and bound algorithm.

The Pseudo-code of the proposed Branch-and-bound algorithm is shown in Fig. 1.

AN ILLUSTRATIVE NUMERICAL EXAMPLE

In this part we present a numerical example for the proposed problem. We consider an instance with 5 non-dummy activities, 4 renewable resources and 2 nonrenewable resources. Figure 2 depicts the network of the problem. The unit costs of nonrenewable resources are 19 and 16 respectively. Other problem data related to renewable resources and activities have been presented in Table 1 and 2.

For this numerical example, we have to schedule dummy activity zero first in AL. This activity is the only activity without any predecessors and it has only

Table 1: Renewable resources data

Renewable resource	Availability	Release time	Due date	Tardiness penalty cost per unit of time
1	4	1	7	8
2	6	0	8	8
3	5	4	11	7
4	5	3	8	8

Table 2: Project activities data

Activity	Number of modes	Mode number	Duration	Renewable resources requirements	Nonrenewable resources requirements
0	1	1	0	0,0,0,0	0,0
1	3	1	8	0,0,0,0	2,2
1	3	2	11	2,0,0,0	2,0
1	3	3	8	0,0,0,0	2,1
2	2	1	3	3,0,0,0	0,0
2	2	2	2	0,0,0,0	0,0
3	1	1	1	2,0,0,0	2,0
4	3	1	9	0,0,0,0	0,0
4	3	2	13	3,0,0,0	2,1
4	3	3	9	0,0,0,4	1,1
5	2	1	3	0,1,0,0	2,0
5	2	2	10	1,0,0,4	1,2
6	1	1	0	0,0,0,0	0,0

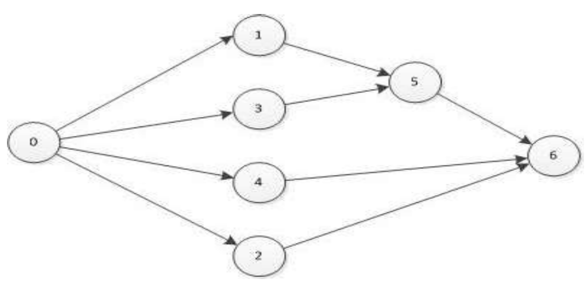

Fig. 2: Project network of the numerical example

one execution mode. So we have only one node in the zero level of the branching tree. Subsequently, in the first level of the tree, four activities 1, 2, 3 and 4 can be placed at the end of current partial AL and as they have 9 different execution modes in total, we have 9 nodes in the first level of the tree.

The activities 0, 3 and 1 listed in the current partial AL have been assigned modes 1, 1 and 2 respectively. In order to determine the lower bound on the nonrenewable resources cost, we choose modes with the least related costs for the rest of activities not listed in current partial AL. As an example, if we choose mode 1 for activity 5 that has 2 different modes, the amount of nonrenewable resources is 2 and 0 with unit cost of 19 and 16 respectively. Therefore the total related cost is 38. Alternatively, if we choose mode 2 for this activity, the amount of usage would be 1 and 2 respectively resulting in the total cost of 51. So for this activity mode 1 is assigned during the process of nonrenewable resources cost lower bound determination.

In order to determine the lower bound of renewable resources cost for the node assumed, first we schedule activities of partial AL. The start times of the three activities 0, 1 and 3 are 0 and their finish times are 0, 11 and 1 respectively. Now we determine the earliest finish time of activity 5. As the predecessors of activity 5 are activities 1 and 3, the earliest start time of activity 5 cannot be less than 11. If we choose mode 1 for this activity, it only needs resource 1 and as the release time of this resource is zero, the earliest start time can be zero which is not feasible according to the possible start time of 11. If we choose mode 2 for this activity, it needs resources 0 and 3 and as their release time are 1 and 3 respectively, the earliest start time is 3 which is again not feasible. Therefore, the earliest start time of 11 is feasible for activity 5. Considering this start time, the finish time of the activity under two modes of 1 and 2 is calculated 14 and 21 respectively, so mode 1 for the activity is assigned and the earliest start time and finish time is set to 11 and 14 respectively.

Once the earliest finish times of activities are determined, we can find the latest possible time at which any renewable resource is needed. In order to find the length of time for renewable execution, for

Table 3: The performance of the branch and bound algorithm

Sample problems sets	# problems getting better results by B&B	# problems solved completely by B&B
J10 (536 problems)	228	536
J16 (550 problems)	249	550
J20 (554 problems)	256	554
J30 (640 problems)	124	529
J60 (640 problems)	87	232
J90 (640 problems)	34	106
J30-r4-n4 (640 problems)	112	498
J60-r4-n4 (640 problems)	71	183

instance activity 5, no resource is needed until the earliest finish time of this activity, because for any of renewable resources this activity has a mode under which it does not need any amount of that resource for execution. Instead, resource 0 is needed for activity 3 at least until the earliest finish time of this activity, because this activity has only one mode under which it needs this resource for execution.

EXPERIMENTAL ANALYSES

In this part we present comprehensive experimental analysis regarding to the problem in question and its developed algorithm. All programs have been coded and executed on C#.NET 2008 platform on a PC with Core 2 Duo 2.53 GHz CPU and 3 GB RAM.

In order to have a full factorial design of parameters, we use sample problems library of PSPLIB [7]. Four sets of multimode project scheduling problems, j10, j16, j20 and j30 have been used as small size and medium size problems. In addition, two sets of large size problems, j60 and j90, have been generated with the same parameters as j30. Also, in order to observe the effect of having more resources in the problem, two extra sets of project scheduling problems have been generated using Progen [6], which we call j30_r4_n4 and j60_r4_n4. All parameters of generation in these sets are similar to the sets j30 and j60 respectively, but instead of having 2 renewable and 2 nonrenewable resources, there exist 4 resources of each type in problems of these sets.

Extra parameters for the instances, which are specific to our MRCPSP-TWRTPC problem and do not exist in simple MRCPSP, have been generated using uniform distribution as following: nonrenewable resources unit usage cost from the range (2, 6), renewable resources tardiness penalty cost from the range (10, 30), renewable resources release time from the range (0, 15) and renewable resources deadline from the range (5, 15) plus the amount of their release time.

Table 3 shows the results of assessing the proposed branch and bound algorithm under CPU-time limitation of 100 seconds for the execution of the branch and bound algorithm. If the process is not finished within the limitation of 100 seconds, the best upper bound gained is reported. A tight initial upper bound for the objective function in branch and bound algorithm is effective in fathoming nodes of branching tree and speeding up the process dramatically. Hence, we used the objective function value of the best solution achieved by a meta-heuristic algorithm for each problem as an initial upper bound for the branch and bound algorithm.

High number of problems in each test function set has been solved to optimality by this algorithm within the CPU-time limit. Forth column of the Table 3 shows the number of problems in each function set that has been solved to optimality by our branch and bound algorithm in maximum CPU-time limit of 100 seconds.

CONCLUSIONS AND FINAL REMARKS

In this paper we introduced MRCPSP-TWRTPC problem as a resource oriented cost minimization project scheduling problem which is based on some rather more extensive realistic assumptions than the similar project scheduling problems have been introduced so far. We formulated this problem as a mixed integer programming model and proposed an exact branch and bound based approach to solve this problem. In order to limit the domain of the branch and bound tree, we suggested to first gain the tight upper bounds by an efficient meta-heuristic approach. Considering the complexity of the presented problem, the computational tests demonstrate the efficiency and effectiveness of our proposed exact method. The results show that our branch and bound approach is able to find the optimal solutions in a short CPU-time for small size of j10 and j16 and medium size of j20 test problems. Also the outcomes express that our exact method can obtain the optimal solutions for large proportion of j30 medium-sized problem sets in a limited CPU-time.

An interesting research topic that might be followed in the future studies is modification and improvement of this proposed branch and bound approach. Also, development of other exact, heuristic or meta-heuristic methods for the problem described in this paper can be an attractive research direction.

REFERENCES

1. Blazewicz, J., J. Lenstra and A. Rinnooy, 1983. Scheduling subject to resource constraints: Classification and complexity. Discrete Applied Mathematics, 5: 11-24.
2. Brucker, P., A. Drexel, R. Mohring, K. Neumann and E. Pesch, 1999. Resource-constrained project scheduling: Notation, classification, models and methods. European Journal of Operational Research, 113: 3-41.
3. Demeulemeester, E. and W.S. Herroelen, 2002. Project Scheduling, A research handbook, Kluwer Academic Publishers.
4. Hartman, S. and D. Briskon, 2011. A survey of variants and extensions of the resource-constrained project scheduling problem. Europan Journal of Operational Research, 207: 1-14.
5. Heilmann, R., 2003. A branch and bound procedure for multi-mode resource-constrained project scheduling problem with minimum and maximum time lags. European Journal of Operational Research, 144: 348-365.
6. Kolisch, R., A. Sprecher and A. Drexl, 1995. Characterization and generation of a general class of resource-constrained project scheduling problems, Management Science, 41: 1693-1703.
7. Kolisch, R. and A. Sprecher, 1996. PSPLIB-A project scheduling problem library European Journal of Operational Research, 96: 205-216.
8. Lova, A., P. Tormos and F. Barber, 2006. Multimode resource-constrained project scheduling: Scheduling schemes, priority rules and mode selection rules. Inteligenica Artificial, 10 (30): 69-86.
9. Lova, A., P. Tormos, M. Cervantes and F. Barber, 2009. An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes. International Journal of Production Economics, 117: 302-316.
10. Peteghem, V. and M. Vanhoucke, 2010. A genetic algorithm for preemptive and non-preemptive multi-mode resource-constrained project scheduling problem. European Joumal of Operational Research, 201: 409-418.
11. Project Scheduling Problem Library-PSPLIB: http://www.129.187.106.231/psplib/.
12. Ranjbar, M., B. De Reyck and F. Kianfar, 2008. A hybrid scatter-search for the discrete time/resource trade-off problem in project scheduling. European Journal of Operational Research, 193 (1): 35-48.
13. Ranjbar, M., M. Khalilzadeh, F. Kianfar and K. Etminani, 2012. An optimal procedure for minimizing total weighted tardiness penalty costs in the resource constrained project scheduling problem. Computers & Industrial Engineering, Article in Press.
14. Sprecher, A., S. Hartman and A. Drexl, 1997. An exact algorithm for project scheduling with multiple modes. OR Spektrum, 19: 195-203.
15. Weglarz, J., J. Jozefowska, M. Mika and G. Waligora, 2010. Project scheduling with finite or infinite number of activity processing modes-A survey. European Journal of Operational Research, Article in Press.
16. Zhu, G., J.F. Bard and G. Yu, 2006. A branch and cut procedure for the multi-mode resource-constrained project scheduling problem. INFORMS Journal on Computing, 18 (3): 377-390.